통보전달대면부(MPI)
함수설명서

국가과학원 콤퓨터과학연구소
우리식조작체계개발조

주체 98(2009)년 4 월

이 문서에서는 MPI 병렬프로그람을 작성할 때 리용되는 MPI 함수들의 사용법과 기능에 대하여 서술한다.

1. MPI_Abort

MPI 실행환경을 끝낸다.

문법

```
#include   "mpi.h"
int  MPI_Abort(MPI_Comm comm, int errorcode)
```

입구파라메터

comm : 끝내려는 과제들이 속해있는 통신기

errorcode : 가동환경에 돌려주는 오유코드

설명

통신기 comm 에 속해있는 모든 MPI 프로쎄스들을 끝낸다. 출구파라메터는 가지지 않는다.

2. MPI_Address

기억기에서의 위치주소를 얻는다.

문법

```
#include   "mpi.h"
int  MPI_Address(void *location, MPI_Aint *address)
```

입구파라메터

location : 호출자기억기에서 위치

출구파라메터

address : 위치의 주소

설명

이 함수는 포트란과 C 프로그람작성자들에게 둘다 제공된다. 이 루틴이 돌려주는 주소는 C에서 &연산자에 의해 나오는 값과 같다.

3. MPI_Allgather

모든 과제들로부터 자료를 모아서 모두에게 전달한다.

문법

```
#include "mpi.h"
int MPI_Allgather(void *sendbuf, int sendcount, MPI_Datatype sendtype, void
      *recvbuf, int recvcount, MPI_Datatype recvtype, MPI_Comm comm)
```

입구파라메터

sendbuf : 송신완충기의 시작주소

sendcount : 송신완충기에 있는 송신될 요소수(옹근수)

sendtype : 송신완충기에 있는 요소의 자료형

recvcount : 임의의 프로쎄스로부터 수신되는 요소개수

recvtype : 수신완충기에 있는 요소의 자료형

comm : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

설명

프로세스 j 에서 보내는 자료블로크를 모든 프로쎄스가 수신하며 수신된 블로크는 완충기 recvbuf 의 j 번째 블로크에 놓인다.

4. MPI_Allgatherv

모든 과제들로부터 자료들을 수집해서 모두에게 전달한다.

문법

#include "mpi.h"

int MPI_Allgatherv (void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int *recvcounts, int *displs, MPI_Datatype recvtype, MPI_Comm comm)

입구파라메터

sendbuf : 송신완충기의 시작주소

sendcount : 송신완충기에 있는 송신될 요소들의 개수(옹근수)

sendtype : 송신완충기에 있는 요소들의 자료형

recvcounts : 매 프로쎄스로부터 수신될 요소수를 담고있는 옹근수배렬(크기는 그룹크기)

displs : 옹근수배렬(크기는 그룹크기), 이 배렬의 i 째 기입은 프로쎄스 i 에서 오는 자료가 놓이게 되는 변위(recvbuf 에 상대적인)를 지정한다.

recvtype : 수신완충기요소의 자료형

comm. : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

설명

프로세스 j 에서 송신한 자료블로크를 모든 프로세스에서 수신하며 수신된 블로크는 완충기 recvbuf 의 j 번째 블로크에 놓인다. MPI_Allgather()함수와 차이점은 매개 프로세스에서 보내는 자료의 개수(블로크크기)가 균일하지 않다는것이다.

5. MPI_Allreduce

모든 프로세스들에 있는 값들을 어떤 연산을 하면서 모으고 그 결과를 모든 프로세스에 배포한다.

문법

```
#include  "mpi.h"
int MPI_Allreduce(void *sendbuf, int sendcount,  MPI_Datatype datatype, MPI_Op
    op, MPI_Comm comm)
```

입구파라메터

sendbuf : 송신완충기의 시작주소
sendcount : 송신완충기에 있는 송신될 요소의 개수(옹근수)
datatype : 송신완충기에 있는 요소의 자료형
op : 수행되여야 할 연산
comm. : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

6. MPI_Alltoall

모든 프로세스로부터 모든 프로세스에로 자료를 보낸다.

문법

```
#include  "mpi.h"
```

int MPI_Alltoall(void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, MPI_Comm comm)

입구파라메터

sendbuf : 송신완충기의 시작주소

sendcount : 송신완충기에 있는 요소의 개수(옹근수)

sendtype : 송신완충기에 있는 요소의 자료형

recvcount : 수신완충기에 수신될 요소의 개수

recvtype : 수신완충기요소의 자료형

comm. : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

7. MPI_Alltoallv

모든 프로쎄스로부터 모든 프로쎄스에로 자료를 보낸다.

문법

#include "mpi.h"

int MPI_Alltoallv(void *sendbuf, int *sendcounts, int *sdispls, MPI_Datatype sendtype, void *recvbuf, int *recvcounts, int *rdispls, MPI_Datatype recvtype, MPI_Comm comm)

입구파라메터

sendbuf : 송신완충기의 시작주소

sendcounts : 매 프로쎄스로부터 송신되는 요소의 개수를 서술하는 옹근수배렬 (크기는 그룹크기)

sdispls : 옹근수배렬(크기는 그룹크기), j 째 항목은 j 째 프로쎄스에로 보낼 자료의 변위(sendbuf 에 상대적이다.)를 지정

sendtype : 송신완충기요소의 자료형

recvcounts : 모든 프로쎄스들로부터 수신될 요소들의 개수를 서술하는 옹근수 배렬(크기는 그룹크기)

rdispls : 그룹크기만한 길이를 가지는 옹근수배렬

recvtype : 수신완충기요소의 자료형

comm : 통신기

출구파라메터

recvbuf : 수신완충기의 시작주소

8. MPI_Attr_delete

지정된 열쇠와 관련되는 속성값의 삭제

문법

```
#include  "mpi.h"
int MPI_Attr_delete( MPI_Comm comm, int  keyval)
```

입구파라메터

comm : 속성이 첨부되여있는 통신기

keyval : 삭제하려는 속성의 열쇠값 (옹근수)

출구파라메터는 없다.

9. MPI_Attr_get

열쇠에 의한 속성값 얻기

문법

```
#include  "mpi.h"
int MPI_Attr_get(MPI_Comm comm, int  keyval, void *attr_value, int *flag )
```

입구파라메터

comm　: 속성이 첨부되여있는 통신기

keyval　: 열쇠값(옹근수)

출구파라메터

attr_value : flag 가 참인 경우 속성값

flag　　: 속성값이 추출되면 참이고 열쇠와 관련된 속성이 없으면 거짓

설명

속성은 MPI_Attr_put()를 리용하여 삽입할 때와 꼭같은 언어로 추출되여야 한다. C 에서는 attr_value 항목이 void*로 선언된다 할지라도 실제로는 void 지적자의 주소이다.

10. MPI_Attr_put

열쇠와 관련되는 속성값을 보관한다.

문법

```
#include "mpi.h"
int MPI_Attr_put(MPI_Comm comm, int keyval, void *attr_value)
```

입구파라메터

comm　　: 속성값이 붙여지는 통신기

keyval　: MPI_Keyval_create()의 귀환값과 같은 열쇠값

attr_value : 속성값

설명

MPI_TAG_UB, MPI_HOST, MPI_IO, MPI_WTIME_IS_GLOBAL 와 같은 영구적인 속성들의 값은 변경시킬수 없다.

속성값의 형은 C 가 리용되고있는가 혹은 포트란이 리용되고있는가에 의존한다. C 에서 속성값은 어떤 지적자이다(void *). 포트란에서는 어떤 단일한 옹근수이다. 포트란은 지적자들을 가지지 않는다. 실례로 32bit 보다 큰 임의의 주소체계는 포트란 DOUBLE PRECISION 에 대하여 64bit 를 리용한다. 만일 속성이 이미 존재한다면 삭제함수(대응하는 keyval 이 창조되였을 때 지정되였던 함수)가 호출된다. 이 함수는 출구파라메터를 가지지 않는다.

11. MPI_Barrier

모든 프로쎄스들을 동기화한다.

문법

```
#include  "mpi.h"
int MPI_Barrier(MPI_Comm comm)
```

입구파라메터

comm : 통신기

설명

모든 그룹성원들이 MPI_Barrier 를 호출할 때까지 호출자(이 함수를 호출한 프로쎄스)의 실행을 중지시킨다. 이 함수는 출구파라메터를 가지지 않는다.

12. MPI_Bcast

”root”프로쎄스로부터 그룹내의 다른 모든 프로쎄스들에 통보를 방송한다.

문법

```
#include  "mpi.h"
int MPI_Bcast(void *buffer, int count, MPI_Datatype, int root, MPI_Comm comm)
```

입구파라메터

buffer : 송신완충기의 시작주소

count : 완충기의 요소개수

datatype : 완충기요소의 자료형

root : 방송하는 프로쎄스의 번호

comm : 통신기

출구파라메터

buffer : 수신완충기의 시작주소

설명

buffer 는 root 에서는 입구파라메터로 다른 모든 프로쎄스들에서는 출구파라메터로 된다.

13. MPI_Bsend

dest 로 주어진 프로쎄스에 사용자가 정의한 완충기의 자료를 보낸다.

문법

```
#include "mpi.h"
int MPI_Bsend(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
      MPI_Comm comm)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기의 요소자료의 크기

datatype : 매 송신완충기의 요소자료의 형

dest : 목적프로쎄스번호

tag : 통보표적

comm : 통신기

설명

이 송신함수는 리용자가 통보문들이 어디에 완충되는가를 알 필요가 없이(리용자는 MPI_Buffer_attach()를 가지고 완충기공간을 제공하여야 하기때문에) 송신하는데 편리하다. 출구파라메터는 없다.

C 에서는

```
MPI_Buffer_detach( &b, &n );
MPI_Buffer_attach( b, n );
```

를 리용하여 통보문을 전송할수 있다.

(MPI_Buffer_detach 는 완충된 모든 통보들이 넘겨질 때까지 끝나지 않는다.)

14. MPI_Bsend_init

완충송신을 위한 손잡이를 구축

문법

```
#include   "mpi.h"
int MPI_Bsend_init(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
      MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신요소의 개수

datatype : 매 요소의 자료형

dest : 목적프로세스의 번호

tag : 통보표적

comm : 통신기

출구파라메터

request : 통신요구

15. MPI_Buffer_attach

사용자가 정의한 송신완충기를 붙인다.

문법

```
#include   "mpi.h"
int MPI_Buffer_attach(void *buffer, int size)
```

입구파라메터

buffer : 완충기의 시작주소

size : 완충기의 크기

16. MPI_Buffer_detach

MPI_Bsend 등에서 리용하기 위하여 현재 완충기를 제거한다.

문법

```
#include   "mpi.h"
int MPI_Buffer_detach(void *bufferptr, int size)
```

출구파라메터

bufferptr : 완충기의 시작주소

size : 완충기의 크기

17. MPI_Cancel

통신요구를 취소한다. 이전에 시작되였던 통신을 취소시킬 필요가 있을 때 리용한다.

문법

```
#include   "mpi.h"
int MPI_Cancel(MPI_Request *request)
```

입구파라메터

request : 통신요구

설명

MPI_Cancel 함수는 출구파라메터를 가지지 않으며 오직 수신요구들에 대해서만 실현되였다. 즉 송신요구들에 대해서는 조작이 없다.

MPI_Cancel 사용에서 기본은 다중완충화수법들인데 여기서는 순수 리론적인 MPI_Irecv 들이 만들어진다. 계산이 끝날 때 이러한 수신 요구들의 일부가 남을수도 있다. 즉 MPI_Cancel 을 리용하여 사용자들이 이런 불필요한 요구들을 취소하도록 한다.

송신조작을 취소하는것은 훨씬 더 어렵다. 기본원인은 송신이 보통 개별적으로 끝나기때문이다. 표적, 크기, 원천프로쎄스에 대한 정보는 보통 즉시에 목적지에 송신된다. 리용자들은 송신취소가 어떤 국부조작(MPI 표준에 의해 정의된것)이 진행되는 동안은 시간이 많이 든다는것을 고려하여야 한다(보통 하나이상의 내부통보문들을 생성하면서).

이 함수는 출구파라메터를 가지지 않는다.

18. MPI_Cart_coords

그룹안에서의 번호를 주면 데카르트위상에서 프로쎄스의 자리표를 결정한다.

문법

```
#include "mpi.h"
int MPI_Cart_coords(MPI_Comm comm, int rank, int maxdims, int *coords)
```

입구파라메터

comm : 데카르트구조를 가진 통신기

rank : 그룹 comm 안에서의 프로쎄스번호

maxdims : 호출하는 프로그람에서 벡토르자리표의 길이

출구파라메터

coords : 지정된 프로쎄스의 데카르트자리표를 포함하는 ndims 크기의 옹근수배렬

19. MPI_Cart_create

위상정보가 첨부된 새로운 통신기를 만든다.

문법

```
#include  "mpi.h"
int MPI_Cart_create(MPI_Comm comm_old, int ndims, int dims, int *periods, int
      *reorder, MPI_Comm, *comm_cart)
```

입구파라메터

comm_old : 입구통신기

ndims : 데카르트살창의 차원수

dims : 매 차원에 있는 모든 프로쎄스개수를 지정하는 ndims 크기의 옹근수배렬

priods : 살창이 주기적인가 아닌가를 지정하는 옹근수배렬
(ndim 크기의 론리값배렬)

reorder : 번호달기가 재순서화되는가(참) 아닌가(거짓)를 지정하는 론리값

출구파라메터

comm_cart : 새로운 데카르트위상을 가진 통신기

20. MPI_Cart_get

통신기와 관련되는 데카르트위상정보를 복귀한다.

문법

```
#include  "mpi.h"
int MPI_Cart_get(MPI_Comm comm, int  maxdims, int *dims, int *periods, int
      *cords)
```

입구파라메터

comm : 데카르트구조를 가진 통신기

maxdims : 호출하는 프로그람에서 벡토르 dims, periods, coords 의 길이

출구파라메터

dims - 매개 데카르트차원에 대한 프로쎄스들의 개수(옹근수배렬)

periods - 매개 데카르트차원에 대한 주기성(참/거짓)(론리값들의 배렬)

coords - 데카르트구조안에서 호출되는 프로쎄스의 자리표들(옹근수배렬)

21. MPI_Comm_compare

두 통신기를 비교한다.

문법

```
#include "mpi.h"
int MPI_Comm_compare(MPI_Comm comm1, MPI_Comm comm2, int *result)
```

입구파라메터

comm1 : 통신기 1

comm2 : 통신기 2

출구파라메터

result : 만일 문맥과 그룹이 다 같으면 MPI_IDENT, 문맥은 다르고 그룹이 같다면 MPI_CONGRUENT, 다른 문맥이지만 류사한 그룹이면 MPI_SIMILAR , 기타 경우에는 MPI_UNEQUAL 인 옹근수

오유

오유처리기는 MPI_Errhandler_set()를 리용하여 변경시킬수도 있으며 미리 정의되여 있는 오유처리기인 MPI_ERRORS_RETURN 에 의하여 귀환되는 오유값들이 생성될수 있다.

MPI_SUCCESS

오유없음, MPI 함수들이 성공적으로 완료되였다.

MPI_ERR_TOPOLOGY

무효한 위상, 이 통신기와 관련되는 위상이 없든가 혹은 정확한 형이 아니다(실례로 MPI_GRAPH 가 기대되고있는데 MPI_CART 가 입력되는 경우).

MPI_ERR_ARG

오유클라스에 의해서 식별되지 않는다(실례로 MPI_ERR_RANK).

22. MPI_Comm_create

새로운 통신기를 창조한다.

문법

```
#include  "mpi.h"
int MPI_Comm_create(MPI_Comm comm, MPI_Group group, MPI_Comm *comm_out)
```

입구파라메터

comm : 통신기

group : 통신기그룹의 부분모임인 그룹(손잡이)

출구파라메터

comm_out : 새로운 통신기(조종기)

오유

모든 MPI 함수(MPI_TIME 과 MPI_WTICK 는 제외)들은 오유값을 돌려준다. C 에서는 함수의 값으로서, 포트란에서는 마지막파라메터로서 오유값을 돌려준다.

MPI_SUCCESS

오유없음, MPI 함수들이 성공적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기를 획득할수 없을 때 발생한다.

련관되는 함수 : MPI_Comm_free

23. MPI_Comm_dup

모든 캐쉬정보를 가지고 통신기를 복제한다.

문법

```
#include "mpi.h"
int MPI_Comm_dup(MPI_Comm comm, MPI_Comm *newcomm)
```

입구파라메터

comm : 통신기(손잡이)

출구파라메터

newcomm : 새로운 문맥을 가지지만 comm 과 같은 그룹으로 되여있는 통신기

설명

이 루틴은 통신문맥은 새롭지만 입구통신기와 같은 그룹의 프로쎄스들을 포함하는 새로운 통신기를 창조하는데 리용된다. 모든 MPI 통신은 한 통신기(그 그룹의 프로쎄스들과 문맥의 합으로서 지정되는)안에서 진행되기때문에 이 루틴은 프로그람모듈 혹은

서고가 리용하기 위한 개별적 통신기를 창조하는 효과적인 방법을 제공한다. 특히 서고루틴이 그 통신기로서 MPI_COMM_WORLD 를 사용하는 일은 없다. 대신 복제된 사용자전용의 통신기가 항상 리용된다.

이 루틴은 본질적으로 통신기의 사본을 생성하므로 입구통신기에 정의되여있는 임의의 속성들도 다 복사한다. 이때 MPI_Keyval_create 함수의 파라메터 copy_function 에 의하여 지정되는 속성복사함수에 의하여 복사가 진행된다. 특히 이것은 그 통신기관련 그룹의 일부 특성(내부련결위상과 같은것)을 서술하는데서와 통신기들을 사용자에게 다시 돌려주는데 쓸수있다. 즉 이 경우에 속성들은 이 통신기에 대한 련속적인 MPI_Comm_dup 조작들을 추적할수 있다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Comm_free, MPI_Keyval_create, MPI_Attr_set, MPI_Attr_delete

24. MPI_Comm_get_name

통신기의 이름을 귀환한다.

문법

#include "mpi.h"

int MPI_Comm_get_name(MPI_Comm comm, char *namep, int *reslen)

입구파라메터

comm : 이름을 얻으려고 하는 통신기(손잡이)

출구파라메터

namep : 통신기의 이름을 가지고있는 출구값. 적어도 MPI_MAX_NAME_STRING 크기의 배렬이여야 한다.

reslen : 이름문자렬의 길이

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. C 에서는 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

25. MPI_Comm_group

주어진 통신기와 관련되는 그룹에 접근한다.

문법

#include "mpi.h"

int MPI_Commgroup(MPI_Comm comm, MPI_Group *group)

입구파라메터

comm : 통신기

출구파라메터

group : 통신기의 그룹

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

26. MPI_Comm_remote_ group

지정된 외부통신기와 련결되여있는 원격그룹에 접근한다.

문법

```
#include "mpi.h"
```

int MPI_Comm_remote_ group(MPI_Comm comm, MPI_Group *group)

입구파라메터

comm : 통신기(외부통신기)

출구파라메터

group : 통신기의 원격그룹

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

27. MPI_Comm_rank

통신기안에서 호출프로쎄스의 번호를 결정한다.

문법

```
#include "mpi.h"
int MPI_Comm_rank ( MPI_Comm comm, int *rank )
```

입구파라메터

comm : 통신기(손잡이)

출구파라메터

rank : comm 그룹에서 호출프로세스의 번호(옹근수)

포트란에 대한 설명

포트란에서 모든 MPI 루틴들(MPI_WTIME 과 MPI_WTICK 는 제외)은 파라메터목록의 마지막에 추가적인 파라메터 ierr 를 가져야 한다. ierr 는 옹근수이며 C 에서 루틴귀환값과 같은 의미를 가진다. 포트란에서 MPI 루틴들은 부분루틴들이며 call 명령에 의하여 호출된다.

모든 MPI 객체들(실례로 MPI_Datatype, MPI_Comm)은 포트란에서 INTEGER 형이다.

오유

모든 MPI 루틴들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. C 함수들은 함수의 값으로, 포트란루틴들은 마지막 파라메터로 돌려준다. 그 값이 귀환되기전에 현재의 MPI 오유처리기가 호출된다. 암시적으로 이 오유처리기는 MPI 일감을 중지시킨다. 오유처리기는 MPI_Errhandler_set 를 리용하여 변경시킬수도 있으며 미리 정의되여있는 오유처리기인 MPI_ERRORS_RETURN 에 의하여 귀환되는 오유값들이 생성될수도 있다. MPI 프로그람에서 어떤 오유가 발생하는 경우 계속 실행될수 없다.

MPI_SUCCESS

오유가 없는 경우; MPI 루틴이 성공적으로 완료되였음

MPI_ERR_COMM

무효한 통신기, 보통 이 오유는 호출시에 null 통신기를 리용한데 있다.
(MPI_Comm_rank 에서는 허용되지 않는다.)

28. MPI_Comm_remote_size

외부통신기에 련결되여있는 원격그룹의 크기를 결정한다.

문법

```
#include "mpi.h"
int MPI_Comm_remote_size(MPI_Comm comm, int *size)
```

입구파라메터

comm : 통신기(손잡이)

출구파라메터

size : 그룹 comm 안에 있는 프로쎄스의 개수(옹근수)

29. MPI_Comm_set_name

통신기에 이름을 설정한다.

문법

#include “mpi.h”

int MPI_Comm_set_name(MPI_Comm comm, char *name)

입구파라메터

comm : 이름을 설정하려는 통신기(손잡이)

name : 통신기이름

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다. 일반적으로 이 오유는 호출시 빈 통신기를 리용한데 있다 (MPI_Comm_rank 에서는 허용되지 않는다.).

MPI_ERR_ARG

무효한 파라메터, 일부 파라메터가 무효하며 구체적인 오유클라스(실례로 MPI_ERR_RANK)로 식별되지 않는다.

30. MPI_Comm_size

통신기와 관련되는 그룹의 크기를 결정한다.

문법

```
#include “mpi.h”
int MPI_Comm_size(MPI_Comm comm, int *size)
```

입구파라메터

comm : 통신기 (손잡이)

출구파라메터

size : 통신기의 그룹안에 있는 프로세스들의 개수(옹근수)

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(실례로 MPI_ERR_RANK)로 식별되지 않는다.

31. MPI_Comm_split

색과 열쇠에 기초하여 새로운 통신기를 창조한다.

문법

```
#include “mpi.h”
```

int MPI_Comm_split (MPI_Comm comm, int color, int key, MPI_Comm *newcomm)

입구파라메터

comm : 통신기

color : 부분모임을 배정하는 조종값(부아닌 옹근수)

key : 프로쎄스번호를 배정하는 조종값(옹근수)

출구파라메터

newcomm : 새로운 통신기(손잡이)

설명

color 는 반드시 부아닌 옹근수이든가 아니면 MPI_UNDEFINED 로 되여야 한다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Comm_free

32. MPI_Comm_test_inter

comm 이 외부통신기인가를 검사한다.

문법

```
#include  "mpi.h"
int MPI_Comm_test_inter(MPI_Comm comm, int *flag)
```

입구파라메터

comm : 통신기(손잡이)

출구파라메터

flag : 기발(론리값)

33. MPI_Errhandler_create

MPI 형식의 오유처리기를 창조한다.

문법

```
#include "mpi.h"
int MPI_Errhandler_create(MPI_handler_function *function,
      MPI_Errhandler  *errhandler)
```

입구파라메터

function : 사용자가 정의한 오유처리기

출구파라메터

errhandler : MPI 오유처리기

설명

MPI 표준에서는 어떤 실현부든지 출구값(오유처리기)이 간단히 그 함수의 주소로 되게 할수 있다고 서술하였다. 그러나 MPI_Errhandler_free 의 동작은 이것을 불가능하게 만든다. 그것은 이 함수가 파라메터의 값을 MPI_ERRHANDLER_NULL 로 설정할것

을 요구하기때문이다. 다시말하여 실제적인 오유처리기는 그것을 리용하는 모든 통신기들이 해방될 때까지 남아있어야 한다.

34. MPI_Errhandler_free

MPI 형식의 오유처리기를 해방한다.

문법

```
#include "mpi.h"
int MPI_Errhandler_free(MPI_Errhandler *errhandler)
```

입구파라메터

errhandler : MPI 오유처리기(손잡이), 끝날 때 MPI_ERRHANDLER_NULL 로 설정된다.

35. MPI_Errhandler_get

통신기에 대한 오유처리기를 얻는다.

문법

```
#include "mpi.h"
int MPI_Errhandler_get(MPI_Comm comm, MPI_Errhandler *errhandler)
```

입구파라메터

comm : 오유처리기를 얻으려는 통신기(손잡이)

출구파라메터

errhandler : 지정된 통신기와 관련되여있는 MPI 오유처리기(손잡이)

36. MPI_Errhandler_set

통신기에 대한 오유처리기를 설정한다.

문법

```
#include "mpi.h"
int MPI_Errhandler_set(MPI_Comm comm, MPI_Errhandler *errhandler)
```

입구파라메터

comm : 오유처리기를 설정하려고 하는 통신기

errhandler : 새로운 통신기에 대한 MPI 오유처리기

37. MPI_Error_class

오유코드를 해당한 오유클라스로 변환한다.

문법

```
#include "mpi.h"
int MPI_Error_class(int errorcode, int *errorclass)
```

입구파라메터

errorcode : MPI 함수가 귀환하는 오유코드

출구파라메터

errorclass : 오유코드에 해당한 오유클라스

38. MPI_Error_string

주어진 오유코드에 대한 문자렬을 복귀한다.

문법

```
#include "mpi.h"
```

int MPI_Error_string(int errorcode,char *string,int *resultlen)

입구파라메터

errorcode : 어떤 MPI 함수 또는 MPI 오유클라스에 의하여 귀환되는 오유코드

출구파라메터

string : 오유코드에 대응하는 문자렬

resultlen : 문자렬의 길이

설명

오유코드들은 C 에서는 MPI 루틴들에 의하여, 포트란에서는 ierr 파라메터에 의하여 귀환되는 값들이다. 이것들은 MPI_Error_class()함수에 의하여 오유클라스로 변환될수 있다.

39. MPI_Finalize

MPI 실행환경을 끝낸다.

문법

```
#include “mpi.h”
int  MPI_Finalize()
```

설명

모든 프로세스들은 끝나기전에 이 함수를 호출하여야 한다. 이 함수가 호출된 후에 실행중에 있는 프로세스들의 개수는 명확치 않다. 즉 MPI_Finalize 를 호출한 후에는 return rc 외에 더 다른것을 수행하지 않는것이 가장 좋다.

40. MPI_Gather

프로세스들의 그룹으로부터 값을 합쳐서 모아놓는다.

문법

#include “mpi.h”

int MPI_Gather(void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype,root, MPI_Comm comm.)

입구파라메터

sendbuf : 송신완충기의 시작주소 (선택)

sendcount : 송신완충기로부터 송신될 요소의 개수 (옹근수)

sendtype : 송신완충기요소의 자료형 (손잡이)

recvcount : 수신완충기에 수신되는 요소의 개수 (root 에서만 의미가 있음)

recvtype : 수신완충기요소의 자료형(root 에서만 의미가 있음)

root : 수신프로쎄스의 번호(옹근수)

comm : 통신기(손잡이)

출구파라메터

recvbuf : 수신완충기의 주소

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부아닌 옹근수이여야 한다.

MPI_ERR_TYPE

무효한 자료형변수, MPI_Datatype 가 승인되지 않았을수 있다.

(MPI_Type_commit 를 보시오.)

MPI_ERR_BUFFER

무효한 완충기지적자, 보통 필요없는 곳에서 null 완충기가 리용되였을 때이다.

41. MPI_Gatherv

그룹안의 모든 프로쎄스로부터 지정된 위치에 자료를 수집한다.

문법

#include “mpi.h”

int MPI_Gatherv(void *sendbuf, int sendcount, MPI_Datatype sendtype, void recvbuf int recvcounts, int *displs, MPI_Datatype recvtype,int root, MPI_Comm comm)

입구파라메터

sendbuf	: 송신완충기의 시작주소 (선택)
sendcount	: 송신완충기에서 송신될 요소의 개수 (선택)
sendtype	: 송신완충기요소의 자료형 (손잡이)
recvcounts	: 수신완충기의 요소자료의 크기
dispel	: 그룹크기만한 옹근수 배렬
recvtype	: 수신완충기요소의 자료형 (root 에서만 의미가 있음) (손잡이)
root	: 수신프로쎄스의 번호 (옹근수)
comm	: 통신기(손잡이)

출구파라메터

recvbuf : 수신완충기의 주소(root 에서만 의미가 있음) (선택)

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_BUFFER

무효한 완충기지적자, 보통 필요없는 곳에서 null 완충기가 리용되였을 때이다.

42. MPI_Get_count

"웃준위" 요소의 개수를 얻는다.

문법

```
#include "mpi.h"
int MPI_Get_count(MPI_Status *status, MPI_Datatype datatype, int *count)
```

입구파라메터

status : 수신조작의 귀환상태

datatype : 매개 수신완충기요소의 자료형

출구파라메터

count : 수신된 요소의 개수(옹근수)

설명

만일 자료형의 크기가 0 이면 이 루틴은 값이 0 인 count 를 귀환한다. 만일 status 안의 자료량이 자료형 datatype 크기의 정확한 배수가 아니면(count 가 옹근수로 안될수도 있으므로) 대신 MPI_UNDEFINED 값을 가진 count 가 귀환된다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

43. MPI_Get_elements

자료형안에서 기본요소들의 개수를 돌려준다.

문법

```
#include "mpi.h"
int MPI_Get_elements(MPI_Status *status, MPI_Datatype datatype, int *elements)
```

입구파라메터

status : 수신조작의 복귀상태

datatype : 수신조작에 리용된 자료형

출구파라메터

count : 수신요소의 개수 (옹근수)

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

44. MPI_Get_version

MPI 의 판본을 얻는다.

문법

```
#include “mpi.h”
int MPI_Get_version(int *version, int *subversion)
```

출구파라메터

version : MPI 의 판본(1 혹은 2)

subversion : MPI 의 부분판본

설명

MPICH 에 기정으로 정의되여있는 값들인 MPI_VERSION 와 MPI_SUBVERSION 는 꼭같은 정보를 포함한다. 이 함수는 서고가 mpi.h 와 mpif.h 파일들에 지정되여있는 판본과 일치하는가를 확인할수 있게 해준다.

45. MPI_Group_compare

두 그룹을 비교한다.

문법

#include “mpi.h”

int MPI_Group_compare(MPI_Group group1, MPI_Group group2, int result)

입구파라메터

group1 : 그룹 1 (손잡이)

group2 : 그룹 2 (손잡이)

출구파라메터

result : 두 그룹의 순서와 성원들이 같으면 MPI_IDENT 인 옹근수, 오직 성원들만 같으면 MPI_SIMILAR, 기타경우에는 MPI_UNEOUAL 인 옹근수

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스로 식별되지 않는다(실례로 MPI_ERR_RANK).

46. MPI_Group_difference

두 그룹의 차로 구성되는 새로운 그룹을 만든다.

문법

#include “mpi.h”

int MPI_Group_difference(MPI_Group group1, MPI_Group group2, MPI_Group *newgroup)

입구파라메터

group1 : 첫번째그룹

group2 : 두번째그룹

출구파라메터

newgroup : 새로운 그룹(손잡이)

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Group_free

47. MPI_Group_excl

현재 그룹에서 일부 성원들을 제외하고 재순서화하면서 그룹을 생성한다.

문법

#include “mpi.h”

int MPI_Group_excl(MPI_group group, int n, int *ranks, MPI_ group *newgroup)

입구파라메터

group : 그룹

n : 프로쎄스번호배렬 ranks 의 크기

ranks : 그룹내에서의 프로쎄스번호배렬로서 새로운 그룹에서는 제외되는 프로쎄스들의 번호

출구파라메터

newgroup : 새로운 그룹, 그룹 group 에 정의된 순서를 그대로 보존

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 특별한 오유클라스로 식별되지 않는다(실례로 MPI_ERR_RANK).

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 번호는 0 과 통신기크기-1 사이의 값이여야 한다. 또는 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수도 있다.

련관되는 함수 : MPI_Group_free

48. MPI_Group_free

그룹을 해방한다.

문법

```
#include “mpi.h”
int MPI_Group_free(MPI_group *group)
```

입구파라메터

group : 그룹 (손잡이)

설명

출력결과로 나오는 그룹은 MPI_GROUP_NULL 로 설정된다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

49. MPI_Group_incl

현재 그룹에 있는 일부 성원들을 취하고 재순서화여 그룹을 생성한다.

문법

#include “mpi.h”

int MPI_Group_incl(MPI_group group, int n, int *ranks, MPI_Group *newgroup)

입구파라메터

group : 그룹

n : 프로쎄스번호배렬의 크기

ranks : 새 그룹에 포함되면서 동시에 group 안에도 있는 프로쎄스번호배렬

출구파라메터

newgroup : ranks 에 정의된 순서를 보존하면서 생성된 새로운 그룹(손잡이)

설명

현재는 프로쎄스번호들이 중복되는가 하는 검사를 하지 못한다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

MPI_ERR_RANK

무효한 원천 또는 목적프로세스번호, 프로세스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 또는 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로세스번호는 MPI_ANY_SOURCE 일수도 있다.

련관되는 함수 : MPI_Group_free

50. MPI_Group_intersection

현존 두개 그룹들의 사귐으로 되는 그룹을 생성한다.

문법

```
#include “mpi.h”
int MPI_Group_intersection(MPI_Group group1, MPI_Group group2, MPI_Group
            *newgroup)
```

입구파라메터

group1 : 첫번째그룹

group2 : 두번째그룹

출구파라메터

newgroup : 사귐그룹(손잡이)

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Group_free

51. MPI_Group_ rang_excl

현존 그룹에서 지정된 범위에 있는 프로세스들을 배제하여 그룹을 생성한다.

문법

#include “mpi.h”

int MPI_Group_rang_excl(MPI_Group group, int n, int ranges, MPI_Group *newgroup)

입구파라메터

group : 그룹

n : 배렬 ranks (옹근수)에 있는 요소의 개수

ranges : 3 개의 옹근수조(시작,끝,폭)를 한 요소로 하는 1 차원 옹근수배렬로서 프로세스그룹 group 에는 있지만 출구그룹인 newgroup 에서는 제외되는 프로세스번호들을 지정.

출구파라메터

newgroup : 그룹 group 에서의 순서를 그대로 보존하면서 새롭게 생성된 그룹

설명

현재 배제되여야 할 ranks 의 매개 프로쎄스번호는 그 그룹에서 유효한 프로쎄스 번호이여야 하며 모든 요소들이 서로 구별되여야 한다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 또는 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수도 있다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

련관되는 함수 : MPI_Group_free

52. MPI_Group_range_incl

현존 그룹에서 지정된 범위에 있는 프로쎄스번호들을 선정하여 새그룹을 창조한다.

문법

```
#include "mpi.h"
int MPI_Group_range_incl(MPI_Group group, int n, int ranges, MPI_Group
        *newgroup)
```

입구파라메터

group : 그룹

n : 배렬 ranges 안에 있는 3옹근수조의 개수(옹근수)

ranges : 3개의 옹근수조(시작,끝,폭)를 한 요소로 하는 1차원 옹근수배렬로서 입구그룹 group 에 있고 출구그룹 newgroup 에도 포함되는 프로쎄스번호들을 지정.

출구파라메터

newgroup : ranges 안에서의 순서를 보존하면서 새롭게 생성된 그룹 (손잡이)

설명

현재 이 실현부는 포함시켜야 할 부분들이 그 그룹안에서 유효한 프로쎄스번호들인가 하는 검사를 하지 않는다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

련관되는 함수 : MPI_Group_free

53. MPI_Group_rank

주어진 그룹에서의 프로쎄스번호를 되돌린다.

문법

```
#include "mpi.h"
int MPI_Group_rank(MPI_Group   group, int *rank)
```

입구파라메터

group : 그룹(손잡이)

출구파라메터

rank : 그룹안에서 호출프로쎄스번호, 프로쎄스가 그룹내의 성원(옹근수)이 아니면 MPI_UNDEFINED 값을 가진다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

54. MPI_Group_size

그룹의 크기를 되돌린다.

문법

```
#include “mpi.h”
int MPI_Group_size(MPI_Group group, int *size)
```

입구파라메터

group : 그룹

size : 그룹내의 프로쎄스개수

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

55. MPI_Group_translate_rank

한개 그룹안의 프로쎄스들의 번호를 또 다른 그룹의 프로쎄스번호로 변환한다.

문법

```
#include“mpi.h”
int MPI_Group_translate_ranks(MPI_Group group1 int n, int *ranks1, MPI_Group
            group2, int *ranks2)
```

입구파라메터

group1 : 그룹 1

n : ranks1 과 ranks2 안에 있는 프로쎄스번호의 개수

ranks1 : 그룹 1 에서의 령 혹은 그 이상의 유효한 프로쎄스번호들의 배렬이다.

group2 : 그룹 2

출구파라메터

rank2 : group2 에 있는 대응하는 번호들의 배렬, 대응하는것이 없을 때 MPI_UNDEFINED 값을 가진다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_GROUP

빈 그룹이 함수의 입구파라메터로 지정되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_BUFFER

무효한 완충기 지적자, 보통 필요없는 곳에서 null 완충기가 리용되였을 때이다.

56. MPI_Group_union

두 그룹을 결합하여 새로운 그룹을 창조한다.

문법

```
#include “mpi.h”
int MPI_Group_union(MPI_Group group1, MPI_Group group2,
                    MPI_Group *grroup_out)
```

입구파라메터

group1 : 첫번째 그룹

group2 : 두번째 그룹

출구파라메터

newgroup : 련합된 그룹

련관되는 함수 : MPI_Group_free

57. MPI_Ibsend

비차단완충송신을 시작한다.

문법

```
#include “mpi.h”
int MPI_Ibsend(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
               MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기안의 요소들의 개수

datatype : 송신완충기요소의 자료형

dest : 목적프로세스번호

tag : 통보표적

comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_BUFFER

무효한 완충기 지적자, 보통 필요없는 곳에서 null 완충기가 리용되였을 때이다.

58. MPI_Init

MPI 실행환경을 초기화한다.

문법

```
#include “mpi.h”
int MPI_Init(int *argc, char  ***argv)
```

입구파라메터

argc : 파라메터의 개수에 대한 지적자

argv : 파라메터벡토르에 대한 지적자

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

설명

포트란에서 이 루틴은 오직 오유귀환파라메터만을 가진다(MPI_INIT(ierror)).

포트란과 C 판본의 MPI_Init 는 차이나기때문에 MPI_Init 를 호출하는데서 제한이 있다. 판본은(포트란 혹은 C) main 프로그람과 정합이 되여야 한다. 즉 main 프로그람이 C 로 되여있으면 C 판본의 MPI_Init 가 호출되여야 한다. 만일 main 프로그람이 포트란이면 포트란판본이 호출되여야 한다.

이 루틴이 끝나면 모든 프로쎄스들이 하나의 파라메터목록사본을 가지게 된다. 이것은 MPI 표준에서는 요구되지 않으며 실제적으로 이식가능한 코드들은 그에 의존하지 않는다. 이것은 이 실현부에서 봉사로 제공된다.

MPI 표준에서는 프로그람이 MPI_INIT 전과 혹은 MPI_FINALIZE 후에 무엇을 할수 있는가에 대하여 설명하지 않았다. MPICH 실현부에서는 사용자가 가능한껏 아무것도

하지 않도록 하였다. 특히 파일열기, 표준입구로부터 읽기, 표준출구로 쓰기와 같은 프로그람들의 외적인 상태를 변경시키는 명령들은 쓰지말아야 한다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

59. MPI_Init_thread

MPI 실행환경을 초기화한다.

문법

```
#include "mpi.h"
int MPI_ Init_thread(int *argc, char ***argv, int required, int *provided)
```

입구파라메터

argc : 파라메터의 개수에 대한 지적자

argv : 파라메터벡토르에 대한 지적자

required : 요구되는 쓰레드지원준위

출구파라메터

provided : 제공된 쓰레드지원의 준위

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

60. MPI_Initialized

MPI_Initialized 가 호출되였는가를 검사한다.

출구파라메터

flag : MPI_Init()가 호출되였으면 참 아니면 거짓을 가리킨다.

61. MPI_Intercomm_create

두개의 내부통신기들로부터 하나의 외부통신기를 창조한다.

문법

```
#include "mpi. h"
int MPI_Intercomm_create (MPI_Comm local_comm, int local_leader,
            MPI_Comm peer_comm, int remote_leader, int tag,
             MPI_Comm *comm_out)
```

입구파라메터

local_comm : 국부적인(내부의)통신기

local_leader : local_comm 의 주프로쎄스번호 (흔히 0)

peer_comm : 원격통신기

remote_leader : peer_comm 안의 원격주프로쎄스번호(흔히 0)

tag : 외부통신기구축에 리용될 통보문표적, 만일 여러개의 MPI_Intercomm_create 가 실행되면 그것들은 서로 다른 표적들을 리용한다. 더 엄밀하게는 국부 및 원격주프로쎄스들이 매개 MPI_intercom_create 에 대해서 서로 다른 표적들을 리용하는것을 담보한다.

출구파라메터

comm_out : 창조된 외부통신기

설명

MPI 표준 1.1 에서는 입구로되는 내부통신기들에 대하여 서로 모순되는 두개의 설명을 가지고있다. 하나는 그 개별적 그룹들이 서로 분리되여야 한다는것이고 다른 하나는 지휘자들이 같은 프로쎄스로 될수도 있다는것이다. 후에 그 그룹들은 서로 분리되여야 한다고 결정되였다. 표준에서 이렇게 하였다 하여 이것을 선택해야 하는것은 아니다. 통신기들에 대한 다른 조작들(MPI_Intercomm_merge 와 같은것)은 그룹들이 분리되여 있다고 생각한다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 역시 MPI_ANY_TAG 일수 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

련관되는 함수 : MPI_Intercomm_merge, MPI_Comm_free, MPI_Comm_remote_group, MPI_Comm_remote_size

62. MPI_Intercomm_merge

외부통신기로부터 내부통신기를 창조한다.

문법

```
#include "mpi. h"
int MPI_Intercomm_merge(MPI_Comm comm, int high, MPI_Comm *comm_out)
```

입구파라메터

comm : 외부통신기

high : 새로운 통신기를 창조할 때 comm 안에 있는 두개 내부통신기들의 그룹을 순서화하는데 리용된다.

출구파라메터

comm_out : 창조된 내부통신기

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Intercomm_create, MPI_Comm_free,

63. MPI_Iprobe

통보문에 대한 비차단검사를 진행한다.

문법

#include “mpi. h”

int MPI_Iprobe(int source, int tag, MPI_Comm comm, int *flag, MPI_Status *status)

입구파라메터

source : 원천프로쎄스번호 혹은 MPI_ANY_SOURCE(옹근수)

tag : 표적값 혹은 MPI_ANY_TAG (옹근수)

comm : 통신기

출구파라메터

flag : (론리형)

status : 상태

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

64. MPI_Irecv

비차단수신을 시작한다.

문법

```
#include "mpi. h"
int MPI_Irecv(void *buf, int count, MPI_Datatype datatype, int source,
                    int tag, MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 수신완충기의 시작주소
count : 수신완충기의 요소개수
datatype : 매개 수신완충기요소의 자료형
source : 원천프로쎄스의 번호
tag : 통보표적
comm : 통신기

출구파라메터

request : 통신요구

65. MPI_Irsend

비차단준비송신을 시작한다.

문법

```
#include "mpi. h"
int MPI_Irsend(void *buf, int count, MPI_Datatype datatype, int dest,
               int tag, MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기에 있는 요소의 수

datatype : 매 송신완충기요소의 자료형

dest : 목적프로쎄스의 번호

tag : 통보표적

comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

66. MPI_Isend

비차단송신을 시작한다.

문법

```
#include “mpi.h”
int   MPI_Isend(void *buf, int count, MPI_Datatype datatype, int dest,
            int tag, MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기에 있는 요소의 개수

datatype : 매 송신완충기요소의 자료형

dest : 목적프로쎄스의 번호

tag : 통보표적

comm　　: 통신기

출구파라메터

request　: 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수,　count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

67. MPI_Irsend

비차단준비송신을 시작한다.

문법

```
#include “mpi.h”
int   MPI_Irsend(void *buf, int count, MPI_Datatype datatype, int dest,
                int tag, MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소
count : 송신완충기에 있는 요소의 수
datatype : 매 송신완충기요소의 자료형
dest : 목적프로쎄스의 번호
tag : 통보표적
comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

68. MPI_Issend

비차단동기송신을 시작한다.

문법

```
#include "mpi.h"
int MPI_Issend(void *buf, int count, MPI_Datatype datatype, int dest,
               int tag, MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기에 있는 요소의 수

datatype : 매 송신완충기요소의 자료형

dest　　　: 목적프로쎄스의 번호
tag　　　: 통보표적
comm　　　: 통신기

출구파라메터
request　: 통신요구

오유
모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.
MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS
오유없음, MPI 함수들이 성과적으로 완료되였다.
MPI_ERR_COMM
무효한 통신기이다.
MPI_ERR_COUNT
무효한 count 변수, count 변수는 부가 아니여야 한다.
MPI_ERR_TYPE
무효한 자료형변수(MPI_Type_commit 를 보시오.)
MPI_ERR_TAG
무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.
MPI_ERR_RANK
무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

69. MPI_keyval_create

새로운 속성열쇠를 생성한다.

문법

#include “mpi.h”

int MPI_keyval_create(MPI_Copy_function *copy_fn, MPI_Delete_function *delete_fn, Int *keyval, void *extra_state)

입구파라메터

copy_fn : 열쇠값을 복사하는 역호출함수

delete_fn : 열쇠값을 삭제하는 역호출함수

extra_state : 역호출함수의 상태

출구파라메터

keyval : 후에 접근하게 될 열쇠값(옹근수)

설명

열쇠값들은 대역적이다(임의의 혹은 모든 통신기에서 리용할수 있다.).

C 와 포트란사이에는 약간한 차이가 있는데 copy_fn 이 MPI_Keyval_create 함수를 호출할 때와 꼭같은 언어로 작성될것을 요구한다는점이다. 이것은 대부분 리용자들에게 문제로 되지 않는다. 그러나 같은 프로그람안에서 포트란과 C 를 둘다 리용하는 프로그람작성자들은 반드시 이 규칙을 따라야 한다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

70. MPI_keyval_free

통신기 캐쉬속성에 대한 속성열쇠를 해방한다.

문법

```
#include “mpi.h”
int MPI_keyval_free(int *keyval)
```

입구파라메터

keyval : 옹근수 열쇠값(선택)을 해방한다.

설명

열쇠값은 대역적이다(이것은 임의의 모든 통신기에서 리용될수 있다.).

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

련관되는 함수 : MPI_Keyval_create

71. MPI_NULL_COPY_FN

속성을 복사하지 않게 하는 함수

설명

이 함수의 리용에 대해서는 MPI_keyval_create 의 설명을 보시오.

72. MPI_NULL_DELETE_FN

속성을 삭제하지 않는 함수

입구파라메터

comm : 통신기

keyval : 열쇠값

attr : 속성

extra_state : 리용자함수들을 주기 위한 리용자정의상태

설명

이 함수의 리용에 대해서는 MPI_keyval_create 의 설명을 보시오.

73. MPI_Op_create

사용자가 정의한 집합함수손잡이를 창조한다.

문법

```
#include "mpi.h"
int MPI_Op_create(MPI_User_function *function, int commute, MPI_Op *op)
```

입구파라메터

function : 사용자정의함수

commute : 통신할수 있으면 참, 아니면 거짓

출구파라메터

op : 연산

사용자함수에 대한 설명

사용자함수에 대한 형정의는 다음과 같이 한다.

typedef void (MPI_User_function) (void * a, void * b, int * len, MPI_Datatype *);

여기서 연산은 b[i] = a[i] op b[i]이며 i=0,...,len-1 이다.

MPI 집합연산루틴(실례로 MPI_Reduce, MPI_Allreduce, MPI_Scan, MPI_Reduce_scatter)에 주어지는 자료형에 대한 지적자는 사용자가 정의한 함수에도 넘겨진다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Op_free

74. MPI_Op_free

사용자가 정의한 집합함수손잡이를 해방한다.

문법

```
#include “mpi.h”
int MPI_Op_free( MPI_Op *op)
```

입구파라메터

op : 연산

설명

이 루틴이 끝날 때 op 는 MPI_OP_NULL 로 설정된다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

련관되는 함수 : MPI_Op_create

75. MPI_pack

자료형을 련속적인 기억기에로 묶기한다.

문법

```
#include “mpi.h”
int MPI_pack(void *inbuf, int incount, MPI_Datatype datatype,
                    void *outbuf, in outcount,int *position, MPI_Comm comm)
```

입구파라메터

inbuf : 입구완충기의 시작주소

incount : 입구자료항목의 개수

datatype : 입구자료의 자료형

outcount : 바이트단위의 출구완충기크기

position : 바이트단위의 완충기에서의 현위치

comm : 묶기되는 통보와 관련되는 통신기

출구파라메터

output : 출구완충기의 시작주소

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(실례로 MPI_ERR_RANK) 로 식별되지 않는다.

련관되는 함수 : MPI_Unpack, MPI_pack_size

76. MPI_pack_size

통보문을 묶는데 필요한 공간의 웃한계를 돌려준다.

문법

```
#include "mpi.h"
int MPI_pack_size(int incount, MPI_Datatype datatype, MPI_Comm comm, int *size)
```

입구파라메터

incount : 묶기함수에 넘기는 개수

datatype : 묶기함수에 넘기는 자료형

comm : 묶기함수에 넘기는 통신기

출구파라메터

size : 묶기되는 통보문크기(바이트단위)의 웃한계 (옹근수)

설명

MPI 표준은 MPI_Pack 에 한하여 이것을 서술하지만 MPI_pack 와 MPI_unpack 에 대하여 둘다 적용된다. 즉 값의 크기는 MPI_pack 나 MPI_unpack 에 의하여 요구되는 최대값이다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

77. MPI_Probe

통보문에 대한 차단검사를 진행한다.

문법

```
#include "mpi.h"
int MPI_probe(int source, int tag, MPI_Comm comm, MPI_Status *status)
```

입구파라메터

source : 원천프로쎄스번호 혹은 MPI_ANY_SOURCE

tag : 표적값 혹은 MPI_ANY_TAG

comm : 통신기

출구파라메터

status : 상태객체

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

78. MPI_Recv

원천프로쎄스에서 보내온 자료를 수신완충기에 넣는다.

문법

```
#include “mpi.h”
int MPI_Recv(void *buf, int count, MPI_Datatype datatype, int source,
      int tag, MPI_Comm comm, MPI_Status *status)
```

입구파라메터

count : 수신완충기의 최대요소수(옹근수)

datatype : 수신완충기요소의 자료형

source : 원천프로쎄스번호

tag : 통보표적

comm : 통신기

출구파라메터

buf : 수신완충기의 시작주소

status : 상태객체

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

79. MPI_Recv_init

수신을 위한 손잡이를 구축한다.

문법

```
#include “mpi.h”
int MPI_Recv_init(void *buf, int count, MPI_Datatype datatype, int source, int tag,
    MPI_Comm comm, MPI_Status *status)
```

입구파라메터

buf : 수신완충기의 시작주소
count : 수신하려는 요소개수
datatype : 매 요소의 자료형
source : 원천프로쎄스번호 혹은 MPI_ANY_SOURCE
tag : 통보표적 혹은 MPI_ANY_TAG
comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_start, MPI_Request_free

80. MPI_Reduce

모든 프로쎄스에 있는 값들을 지정된 연산을 하면서 하나의 값으로 축소한다.

문법

#include “mpi.h”

int MPI_Reduce(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, int root, MPI_Comm comm)

입구파라메터

sendbuf : 송신완충기주소

count : 송신완충기에서의 요소개수

datatype : 송신완충기요소의 자료형

op : 축소연산

root : 뿌리프로쎄스번호

comm : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

알고리듬

이것은 현재 단순한 나무알고리듬을 실현한것이다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

81. MPI_Reduce_scatter

값을 결정하고 결과를 뿌린다.

문법

```
#include "mpi.h"
Int MPI_Reduce_scatter(
                    void *sendbuf,
                    int sendcount,
                    MPI_Datatype sendtype,
                    void *recvbuf,
                    int recvcount,
                    MPI_Datatype recvtype,
                    int root,
                    MPI_Comm comm)
```

입구파라메터

sendbuf   : 수신완충기의 시작주소

sendcount : 매개 프로쎄스에 송신하는 요소개수 (옹근수, root 에서만 의미를 가진다.)

sendtype : 송신완충기요소의 자료형 (root 에서만 의미를 가진다.) (손잡이)

recvcount : 매 프로쎄스에서 수신되는 요소개수 (옹근수)

recvtype : 수신되는 완충기요소의 자료형

root : 자료를 전송하는 프로쎄스의 번호(옹근수)

comm : 통신기

출구파라메터

recvbuf : 수신완충기의 주소

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

82. MPI_Request_free

통신요구객체를 해방한다.

문법

```
#include "mpi.h"
int MPI_Request_free(MPI_Request *request)
```

입구파라메터

request : 통신요구

설명

이 루틴은 보통 MPI_Recv_init 나 MPI_Send_init 또는 그런 류형의 루틴들이 창조한 련속적인 요구들을 해방하는데 리용된다. 그러나 이것은 MPI_Irecv 혹은 MPI_Isend 또는 그런 류형의 루틴들이 창조한 한개 요구를 해방하는데도 리용할수 있다. 이 경우에는 그 요구에 대해서 test 또는 wait 루틴들을 리용하지 못할수 있다.

활성상태에 있는 요구를 해방하는것도 허용된다. 그러나 일단 해방되면 wait 혹은 test 루틴들에서 그 요구를 리용할수 없다 (실례로 MPI_Wait).

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_ARG

무효한 변수, 일부 변수가 무효하며 구체적인 오유클라스(즉 MPI_ERR_RANK)로 식별되지 않는다.

련관되는 함수 : MPI_Isend, MPI_Irecv. MPI_Issend, MPI_Ibsend, MPI_Irsend, MPI_Recv init, MPI_send_init, MPI_Ssend_init, MPI_Rsend_init, MPI_Wait, MPI_Test, MPI_Waitall, MPI_Waitany, MPI_Waitsome, MPI_Testall, MPI_Testany, MPI_Testsome

83. MPI_Rsend

준비송신을 진행한다.

문법

#include “mpi.h”

int MPI_Rsend(void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기에서 요소의 개수

datatype : 매 송신요소의 자료형

dest : 목적프로쎄스의 번호

tag : 통보표적

comm : 통신기

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

84. MPI_Rsend_init

준비송신을 위한 손잡이를 구축한다.

문법

```
#include "mpi.h"
int MPI_Rsend_init(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
            MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신완충기에서 요소의 개수

datatype : 매 요소의 자료형

dest : 목적프로쎄스의 번호

tag : 통보표적

comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Start, MPI_Request_free, MPI_send_init

85. MPI_Scan

집합처리들에서 자료의 조사(부분적인 축소들)를 진행한다.

문법

#include “mpi.h”

int MPI_Scan(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, MPI_ Comm comm)

입구파라메터

sendbuf : 송신완충기의 시작주소

count : 입구완충기에서 요소의 개수

datatype : 입구완충기에 있는 요소의 자료형

op : 연산

comm : 통신기

출구파라메터

recvbuf : 수신완충기의 시작주소

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

86. MPI_Send

목적프로쎄스에로 통보를 보낸다.

문법

```
#include “mpi.h”
int MPI_Send(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
             MPI_Comm comm)
```

입구파라메터

buf : 송신완충기의 시작주소
count : 송신완충기에서 요소의 개수
datatype : 매 송신완충기 요소의 자료형
dest : 목적프로쎄스의 번호
tag : 통보표적
comm : 통신기

설명

이 루틴은 통보가 다 수신될 때까지 그 통보에 대한 접근을 차단시킨다.

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수 count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

련관되는 함수 : MPI_Isend, MPI_bsend

87. MPI_Send init

표준송신을 위한 손잡이를 구축한다.

문법

```
#include “mpi.h”
int MPI_Send_init(void *buf, int count, MPI_Datatype datatype, int dest, int tag,
            MPI_Comm comm, MPI_Request *request)
```

입구파라메터

buf : 송신완충기의 시작주소

count : 송신하려는 요소의 개수

datatype : 매 요소의 자료형

dest : 목적프로쎄스의 번호

tag : 통보표적

comm : 통신기

출구파라메터

request : 통신요구

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.

련관되는 함수 : MPI_Start, MPI_ Start_all, MPI_Rqeust_ free

88. MPI_Sendrecv

통보문을 수신하고 송신한다.

문법

```
#include “mpi.h”
int MPI_Sendrecv(void *sendbuf, int sendcount, MPI_Datatype sendtype,  int dest,
                 int sendtag,
                 void *recvbuf, int recvcount,   MPI_ Datatype sendtype, int source,
                 int recvtag, MPI_ Comm comm, MPI_Status *status)
```

입구파라메터

sendbuf : 송신완충기의 시작주소
sendcount : 송신완충기에 있는 요소개수
sendtype : 송신완충기에 있는 요소의 자료형
dest : 목적프로쎄스의 번호
sendtag : 송신표적
recvcount : 수신완충기의 요소개수
souce : 원천프로쎄스의 번호
recvtag : 수신표적
comm : 통신기

출구파라메터

recvbuf : 수신완충기의 시작주소
status : 상태

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

89. MPI_Sendrecv_replace

하나의 완충기를 사용하여 송신과 수신을 한다.

문법

#include “mpi.h”

int MPI_Sendrecv_replace(void *buf, int count, MPI_Datatype sendtype, int dest, int sendtag, int source, int recvtag, MPI_ Comm comm, MPI_Status *status)

입구파라메터

count : 송수신완충기의 요소개수

datatype : 송수신완충기의 요소형

dest : 목적프로쎄스의 번호

sendtag : 송신통보표적

source : 원천프로쎄스의 번호

recvtag : 수신통보표적

comm : 통신기

출구파라메터

buf : 송수신완충기의 시작주소

status : 상태

오유

모든 MPI 함수들(MPI_Wtime 과 MPI_Wtick 는 제외)은 오유값을 돌려준다. 례하면 C 함수들은 함수의 값으로서, 포트란함수들은 마지막파라메터에 오유값을 되돌린다.

MPI 프로그람은 오유발생시 계속 실행될수 없다.

MPI_SUCCESS

오유없음, MPI 함수들이 성과적으로 완료되였다.

MPI_ERR_COMM

무효한 통신기이다.

MPI_ERR_COUNT

무효한 count 변수, count 변수는 부가 아니여야 한다.

MPI_ERR_TYPE

무효한 자료형변수(MPI_Type_commit 를 보시오.)

MPI_ERR_TAG

무효한 표적인수, 표적들은 부가 아니여야 하며 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 표적들은 MPI_ANY_TAG 로 될수도 있다. 가장 큰 표적값들은 속성 MPI_TAG_UB 를 통하여 얻을수 있다.

MPI_ERR_RANK

무효한 원천 또는 목적프로쎄스번호, 프로쎄스번호는 0 과 통신기크기-1 사이의 값이여야 한다. 수신함수들(MPI_Recv, MPI_Irecv, MPI_Sendrecv 등)에서 프로쎄스번호는 MPI_ANY_SOURCE 일수 있다.

MPI_ERR_INTERN

이 오유는 MPI 실행중에 기억기획득이 불가능할 때 발생한다.